

# SPECIFICATION

Docket No. 0544MH-40022

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN that we, Conor McGann and Bruce Macartney-Filgate, residing in the State of Texas, have invented new and useful improvements in a

## **DISTRIBUTED SESSION SERVICES**

of which the following is a specification:

## CROSS-REFERENCE TO RELATED APPLICATION

1        This application claims the benefit of United States Provisional  
2        Application No. 60/158,733, filed October 11, 1999.

## BACKGROUND OF THE INVENTION

3        1.        Field of the Invention:

4               The present invention relates generally to computer communication  
5        systems, and more specifically to a method and system for managing session level  
6        services over a complex Internet service application.

7        2.        Description of the Prior Art:

8               Use of publicly available communication systems such as the Internet is  
9        currently increasing at a rapid rate. Many companies find it extremely valuable to  
10       be able to provide information to customers and other entities over such  
11       communication systems. In particular, web pages are made publicly available,  
12       and typically accessed by widely available browsers.

13              Because a large number of people may wish to access a company's  
14       information simultaneously, it is becoming more common to provide a bank of  
15       web servers that can operate in parallel. Each web server has access to the

1 underlying corporate information as is well known in the art, and each can handle  
2 a number of sessions simultaneously.

3 Each connection between a remote browser and the underlying system  
4 can be termed a session. A session is, generally, a series of requests from a  
5 browser to a web system that make up an interaction with the characteristics of a  
6 conversation. The system retains data as the interaction continues (known as  
7 session data) that enables the system to remember what has been done in  
8 earlier stages of the interaction. Use of sessions is known to those skilled in the  
9 art. The web hosting system, containing several web servers in parallel,  
10 maintains information on each of these sessions in order to provide appropriate  
11 communications with the remote browser.

12 An example of a system which uses a bank of web servers in this matter is  
13 described in detail in United States Patent No. 6,076,108, issued to *Courts et al.*,  
14 and assigned to i2 Technologies, Inc. The method and system set forth in this  
15 patent utilizes a global session server (GSS), which maintains information on all  
16 of the sessions currently running on the host system. A single session server, or  
17 a bank of such session servers, are used to provide session information to each  
18 workstation so that a proper response can be made to request by remote  
19 browsers.



**THE UNIVERSITY OF CHICAGO**

1 In accordance with the present invention, a remote session server, or bank  
2 of remote session servers, is provided to a bank of web servers. Each web server  
3 maintains a cache that contains all of the session information for all sessions being  
4 serviced by that server. The web server utilizes its local cache to perform all  
5 session services. The remote session server maintains a copy of all session  
6 information from all caches, and is updated whenever changes are made on each  
7 web server. If a web server should go down, an ongoing session can be  
8 transferred to a different web server. In such event, the new web server is able to  
9 obtain the previous status of the session from the remote web server and continues  
10 serving the session in a transparent manner.

## BRIEF DESCRIPTION OF THE DRAWINGS

1           The novel features believed characteristic of the invention are set forth in the  
2 appended claims. The invention itself however, as well as a preferred mode of use,  
3 further objects and advantages thereof, will best be understood by reference to the  
4 following detailed description of an illustrative embodiment when read in  
5 conjunction with the accompanying drawings, wherein:

6           Figure 1 is a block diagram of a system containing a plurality of web servers;

7           Figure 2 is a block diagram of the system of Figure 1 including a remote  
8 session server;

9           Figure 3 is a partial block diagram of one of the web servers in Figures 1 and  
10 2; and

11           Figure 4 is an alternative embodiment of the system in Figure 2 that includes  
12 multiple remote session servers.

## DESCRIPTION OF THE PREFERRED EMBODIMENT

1       The system and method of the present invention is an improvement to the  
2 global session server described in detail in United States Patent No. 6,076,108.  
3 However, it will be understood by those skilled in the art that the described  
4 technique for providing a remote session server can be adapted to any system  
5 using a plurality of web servers to handle Internet or similar communications.

6       The information stored for each session will be implementation dependent,  
7 and is known by those skilled in the art. For example, session data can typically  
8 include information such as the user profile, status of the session, history of events  
9 in the session, and particular information pertinent to the session such as an  
10 identification of items in a shopping cart for purchases being made by the remote  
11 user. Identification and maintenance of such information is well known by those  
12 skilled in the art, and will not be described further herein.

13       Referring to Figure 1, a host system designated generally as reference  
14 number 10 includes a plurality of web servers 12, 14, 16, 18, 20, 22. A browser  
15 connect 24 connects to one of the web servers 12-22 over an internet connection  
16 26 which is routed through a local director 28. As known in the art, local director 28  
17 directs all communications from incoming IP addresses to selected servers 12-22.  
18 This may be through the use of a table, or other algorithm, that distributes  
19 communications with connected IP addresses to particular web servers 12-22.

1 Local director 28 functions to keep a specific browser connected to a specific web  
2 server to optimize access to a session for normal operation.

3 In prior art systems, as exemplified by United States Patent No. 6,076,108,  
4 each web server 12-22 communicates with a separate global session server that  
5 contains all session information. When an event occurs which requires access to  
6 such session information, such as display of a web page requested by a browser  
7 24, access must be made to the global session server. However, in the preferred  
8 technique, each web server 12-22 maintains all of its session service information on  
9 the same hardware system that provides the remaining web services.

10 Referring to Figure 2, a remote session server 30 is provided which  
11 communicates with all of the web servers 12-22. Remote session server 30 is  
12 provided to keep track of, and maintain a backup for, all of the session information  
13 on each of workstations 12-22.

14 Referring to Figure 3, workstation 12 is illustrated in more detail. Contained  
15 within workstation 12 is a session server cache 32, which contains all of the session  
16 ID's (SID) 34 which are being handled by web server 12. For each session ID 34,  
17 all session data 36 which is maintained on behalf of the web server is maintained  
18 within session cache 32. Thus, when a request is made by a particular session,  
19 web server 12-22 contained locally all information needed to respond to the  
20 request. It is not necessary to obtain session information from any remote location.



1           However, whenever session information within cache 32 is updated by any  
2 web server, a copy of such update is communicated to remote session server 30.  
3 Thus, remote session server 30 contains a complete set of all session information  
4 for all sessions on all web servers.

5           If a particular web server should go down, all sessions on that web server  
6 must be handled elsewhere. When a web server goes down, local director 28  
7 becomes aware of that fact and re-routes sessions, based upon IP addresses, to  
8 alternate web servers. When sessions that were connected to the failed web  
9 server make subsequent requests, their session data is retrieved by the alternate  
10 servers from the remote session server. If a web server goes down, changes to the  
11 session will be lost if the web server goes down before those changes have been  
12 updated in the remote server. In that case, the session will be available on the  
13 alternate server, but the session will not be completely current

14           For example, assume web server 16 goes down while running 100  
15 simultaneous sessions. At the time web server 16 goes down, a complete copy of  
16 its session cache 32 was stored on remote session server 30. Some recent  
17 information may be lost if the crash of web server 16 prevents the last few cache  
18 updates being written to remote session server 30, but all sessions should  
19 otherwise be reasonably current.

20           Using whatever algorithm is implemented in local director 28, all of the  
21 sessions that were on web server 16 are re-routed to one of the other web servers

1 the next time a communication is received. In some systems, a standby web  
2 server can be used; in others already operating web servers will take over the load  
3 from the system that went down. In either event, when local director 18 routes an  
4 interrupted session to a new web server, for example web server 12, the web  
5 server is initially unprepared to handle the transferred session.

6 Before responding to the transferred session the first time, web server 12  
7 obtains from remote session server 30 a copy of the session information for the  
8 transferred SID. Once this data is transferred from remote session server 30,  
9 workstation 12 is ready to manage the session from the point at which workstation  
10 16 left it.

11 In the system shown in the Figures, if remote session server 30 goes down,  
12 all of the web servers 12-22 continue functioning. In other words, the remote  
13 session server 30 is not a single failure point that will bring down the entire system.  
14 Once remote session server 30 is brought back up, all of the web servers 12-22,  
15 which have been handling their sessions properly in the meantime, will update all of  
16 their changed sessions to remote session server 30.

17 In the scheme shown in Figure 2, if a web server should go down while  
18 remote session server 30 is unavailable, the current status of all sessions on that  
19 web server will be lost. In order to provide addition robustness to the system,  
20 remote session server 30 can be implemented as a bank of separate machines.

Referring to Figure 4, remote session server 30 is actually composed of two separate pieces of hardware, servers 38, 40. Each workstation 12-22 maintains an identifier indicating which RSS 38, 40 maintains its session data backups. If an RSS 38, 40 used by a particular workstation should go down, that workstation connects to another RSS within the bank making up remote session server 30, and provides it a complete copy of its session information cache 32. In this manner, all sessions are properly handled even if problems should be encountered in one or more web servers, and one or more servers in the remote session server bank 30, simultaneously.

When a workstation goes down, and a session is moved to a new workstation, the new workstation will need to poll the individual machines within the remote session server 30 in order to determine which one has that session data. Once the proper machine is found, the new workstation downloads the session data and continues as described above.

Within each workstation, it is possible for multiple updates to be attempted to the data for a single session. For example, if two frames for a session try to update at the same time, the remote session server 30 is not updated until both updates are made. By locking the data for a particular SID whenever an update is made, multiple updates will be automatically serialized. If an attempt is made to obtain a lock to update session information, and that session is already locked, the update will be retained in a queue until the previous update is completed. Only when all pending updates for a particular SID have been made will the corresponding

1 session information be transferred from the workstation to the remote session  
2 server 30 to provide the remote update.

3 The described system and method provides for a bank of web servers that  
4 provide excellent performance because all of the session information needed by  
5 each server is maintained locally. In addition, by backing up the local information  
6 cache to a remote session server, a session is maintained even if its hosting web  
7 server should go down. By providing a bank of remote session servers, even a  
8 failure of one of the machines in the remote session server bank will not terminate a  
9 session.

10 While the invention has been particularly shown and described with  
11 reference to a preferred embodiment, it will be understood by those skilled in the art  
12 that various changes in form and detail may be made therein without departing from  
13 the spirit and scope of the invention.